APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: A TOOL FOR EXTRACTING AND MANIPULATING COMPONENTS OF WARPING TRANSFORMS

APPLICANT: TODOR G. GEORGIEV

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL485780835US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the U.S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202.

November 28, 2001
Date of Deposit

Signature

LeRoy Jenkir
Typed or Printed Name of Person Signing Certificate

A TOOL FOR EXTRACTING AND MANIPULATING COMPONENTS OF WARPING TRANSFORMS

TECHNICAL FIELD

This invention relates to digital image manipulation, and more particularly to warping.

BACKGROUND

With the advent of digital photography, the desire has grown to be able to manipulate the images produced. A user may wish simply to enhance an image, or to distort it for a visual effect, or to touch-up the image. A variety of computer programs exist (such as Adobe® Photoshop®) designed to aid a user to alter an image. Such alterations can include a variety of distortions, such as a swirling effect, or magnification of certain areas of an image, or stretching portions of an image. Other distortions are common.

A distortion may be thought of as a grid of vectors (the "distortion grid,") with each vector corresponding to a single point in an image. The vectors indicate how the image is modified to obtain the original, undistorted source image. The source image itself may consist of a grid of vectors each with zero length (as well as other information, such as color). The image with the distortion applied is the "destination image."

As distortions are sequentially applied, they are summed in the distortion grid. What exists in the distortion grid is only the sum of all previous distortions, without the history of the individual distortions themselves.

Once the distortion grid is satisfactory to a user (who typically sees the distorted image on a graphical display device while working on it) the user may elect to accept the changes. Once that is done, typically, the distorted image in effect replaces the original, and the information regarding the distortions (i.e. the sum of the distortions) is lost. If the user elects not to accept the distortion, the original image is preserved and the distortion grid discarded.

SUMMARY

In one aspect, the invention includes a method for responding to a user action selecting an area in an image related to an area in a distortion grid, and using two or more points local to the area, extracting one or more components of the local distortion. The method may include

exctracting displacement, rotation, magnification, skew, directional scaling or combinations. In another aspect, the method includes calculating an afine transform from the local points and may also include decomposing the affine transform into one or more components of a warping. The extraction may include calculating a determinant of a resulting linear transform matrix representing a linear transform derived from the affine transform to obtain rotation, or using eigenvalues of the linear transform matrix to obtain scaling. Rotation or skew may be removed prior to extraction of scaling.

In another aspect of the invention, a user can select desired components to be extracted from an on-screen menu, and may further select the area from which to extract the selected component(s) using a virtual paintbrush.

In yet another aspect, the selected component(s) of the warping may be applied by using a virtual paintbrush to the source image or to another image.

In another aspect, the invention includes a computer program product to accomplish the extraction and other aspects described above.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a diagram of a distortion grid and an image grid.

FIG. 2 is a diagram of an original image and a distorted image.

FIG. 3 shows a menu selection screen.

FIG. 4 is a diagram of a portion of a distortion grid.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

A distortion can be viewed as a smooth mapping from the image plane of the source image to the image plane of the destination image. We define this mapping by the distortion grid on the destination image. A distortion can be very complex. However, because it is a smooth mapping, any distortion may be viewed locally as an affine transform, consisting of a translation and a linear transform. The affine transform can be further decomposed into elements or components of a distortion such as translation, magnification, rotation, directional scaling and

skew (also referred to as shear.) It is desirable to perform this decomposition because these components are more intuitive for a user. A user need not know mathematics to understand the effect of a rotation, or what an affine transform is to know the user wants a magnification.

Having created a distortion of an image, a user may wish to use selected components of a distortion in a given area of an image, and apply those selected components elsewhere. The user may wish to keep only the rotation, or the scaling, or to spread the existing local distortion over a wider area. The present invention enables a user to select easily one or more components of a distortion local to a point, and using a virtual paintbrush, apply the selected components elsewhere. ("Local to a point" means adjacent to, or within a few pixels of, the point. The farther away, the less it is local to a point.) Alternatively, the user can select components of a distortion and remove them using the virtual paint brush.

A distortion (the warping transform) may well have resulted from a complex sequence of distortions no one of which corresponds to any of the final decomposition elements. While the local distortion at a point may be decomposed into a specific set of components, it cannot be determined how the distortion at that point was in fact created.

A virtual paintbrush can be implemented with a cursor manipulated by a mouse or other input device, which causes the cursor to be moved about a computer display screen over an image, a virtual "canvas", and thereby applies an effect to the image. The user may typically select from among a variety of brush shapes, e.g., circle, square, or single pixel. The user drags the brush along a "trajectory" from one location to another within the image, thereby defining an "envelope." Pixels within the envelope are affected by the operation of the brush. One type of brush allows a user to "paint" colors and textures into the envelope. The paint brush replaces the pixels in the envelope with the image of the brush. Other effects may also be applied with a brush, such as a warping.

It is desirable that the control of a distortion and the manipulation of an image be intuitive and simple. Adobe Photoshop offers a virtual paintbrush that is movable about an image by a user manipulating a mouse. In an exemplary implementation, a computer program to edit images enables a user to select components of a distortion the user desires to manipulate by checking boxes on a menu. Other selection methods may also be used beneficially. Having selected the desired component(s) the user may then select a point in the distorted image (corresponding to an

element of the distortion grid) by moving a cursor to the desired point, then mouse click over the point to extract the selected component(s).

The selected components may then be applied by moving the virtual paintbrush around the distorted image, adding or subtracting out the selected components. In this implementation, the paintbrush can have a variety of shapes and application functions affecting how the application is effected, as described in greater detail below.

FIG. 1 shows an image 10 with a pixel 12. The image may be thought of as being associated with zero distortion vectors (representing no distortion).

FIG. 1 further shows a distortion grid, of the same size as the image, but having non-zero distortion vectors 14,16,18 in the grid. At each point these vectors represent the sum of the distortions applied to the image.

FIG. 2 shows an original image 30 and a distorted (warped) image 32. The distorted image may have been created by dragging portions of the head and neck upward and to the right, and rotating parts of the body. In fact, it was created by a combination of steps, including stretching, compressing, and swirling portions of the image. It is not so important to know exactly how the distortion was created, but rather to note that the distortion at any point in the image may be viewed locally as a combination of a translation and a linear transform.

Suppose a user wishes to obtain just the rotational component of the distortion at one area of the distorted image. FIG. 3 shows a sample menu selection screen 34 that a user can employ to select just the desired elements of a warping, in this case selecting only the rotation 36. Other selectable elements shown include translation 37 and vertical skew 38. Other menu items are available.

Having selected the desired elements of a distortion, the user can cause the mouse cursor to hover over the area of the distorted image from which the user desires to draw the selected distortion components. Clicking the mouse at that point causes the selected components of the distortion in the area of that point to be extracted.

FIG. 4 shows four elements 40 of a distortion grid. (For ease of visualization, the vectors at each element are not shown.) Suppose the user's selected point is A 42. This point has nearby points B 44, C 46 and D 48 local to point A 42. (Other local points could have been used, such as points above A or to the left of A. Any points which are all local (i.e. within a small area) to the selected point A 42 may be used.

The warping transform at a small area local to a point can be approximated by an affine transform. The affine transform has two parts: displacement and linear transform. The displacement can be taken as the original vector at point A 42. This is similar in effect to choosing the vector at any of the other local points B 44, C 46 or D 48, or more complex approaches such as averaging local vectors, but using a single point is sufficient and faster.

The distortion vector at any local point can be represented as the displacement vector at that point plus a linear transformation of the vector $\vec{V}$ = (u, v) that starts from A 42 and ends at that point. The linear transformation is represented by a 2X2 matrix M_0:

$$\mathbf{M_0} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

and it is the same when calculating all corners A 42, B 44, C 46 and D 48 in the same small local area. Given the distortion vectors at the four points, we find M_0 by solving

EQUATION 1 $\vec{D}$ = (Displacement at point A) + $\mathbf{M_0} \cdot \vec{V}$

for points B 44 and C 46.

The total distortion vector at a point is represented by

$$\vec{D} = \begin{pmatrix} Dx \\ Dy \end{pmatrix}$$

In the exemplary implementation described here,

$$\vec{V} = \begin{matrix} u \\ v \end{matrix}$$

is the vector from point A 42 to the points B 44 or C 46. Point D 48 or other local points can also be used, but doing so can result in more complex calculations, including averaging out among any number of points.

Now, finding the matrix M_0 is easy:

The vector pointing to B 44 is the vector $\vec{V} = (1, 0)$

The vector pointing to C 46 is the vector $\vec{V} = (0, 1)$

(choosing the y-axis pointing down)

To simplify notation, we call the total displacement vector $\vec{D}$ at point A 42 simply $\vec{A}$ with components Ax and Ay, and similarly for the total displacement at B 44 and C 46.

If we substitute those into EQUATION 1 we get:

a = Bx - Ax

b = Cx - Ax

c = By - Ay

d = Cy - Ay

The final result of extracting the Affine Transform is the location (x,y coordinates) of the selected point A 42, the displacement vector at this point and the local matrix M_0.

Having determined the Affine Transform, we may extract from it a new $\mathbf{M_x}$ which has only a selected component of the local distortion, such as rotation or magnification. The selected component matrix is given the subscript m for magnification, r for rotation and so forth.

1. Extract Displacement only:

From equation 1, we need simply imply let $\mathbf{M_0}$=1 and use only the displacement component..

2. Extract magnification only.

Calculate the determinant of $\mathbf{M_0}$:

$$\det = a \cdot d - b \cdot c$$

If the determinant is less than zero, then there is a reflection, which must first be dealt with as discussed below. A determinant of exactly zero results in a single point, hence is not encountered in application.)

If the determinant is greater than zero, we set

$$\mathbf{M_m}=\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix}$$

Where $s = \sqrt{\det}$

3. Extract rotation

If the determinant is less than zero, then there is a reflection, which must first be dealt with as discussed below.

If the determinant is greater than zero: calculate the angle ϕ:

$$\phi = \arctan\left(\frac{b-c}{a+d}\right)$$

Alternatively, φ may be calculated:

$$\phi = \frac{\arctan\left(\frac{-b}{a}\right)+\arctan\left(\frac{-c}{d}\right)}{2}$$

EQUATION 2 $$\mathbf{M_r}=\begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

4. Extract rotation and magnification

$\mathbf{M_{rm}}$ is the product (matrix multiplication) of the rotation matrix $\mathbf{M_r}$ and the magnification matrix $\mathbf{M_m}$.

5. Handling reflection

If the determinant of $\mathbf{M_0}$ is less than zero, we know there is a reflection. We approach this case in a special way:

First we remove the reflection by multiplying $\mathbf{M_0}$ by the inverse reflection matrix $\mathbf{R^{-1}}$

$$\mathbf{R^{-1}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

so instead of $\mathbf{M_0}$ we use $\mathbf{R^{-1}} * \mathbf{M_0}$.

Then we proceed as above for displacement, magnification, rotation, skew (described below) and combinations of components. In the end, the reflection matrix **R** is applied to get the resulting $\mathbf{M_1}$. (Note that $\mathbf{R^{-1}}=\mathbf{R}$)

6. Handling directional scaling

In this case, there must be no rotation, skew or reflection, that is, $\mathbf{M_0}$ must be symmetric. (If any of these components exist, the matrix will not be symmetrical, and they must first be removed, as described above for rotation and below.) Directional scaling is scaling stronger in one direction than in another. Magnification (positive or negative) is equal scaling in all directions.

If $\mathbf{M_0}$ is symmetric, it has real eigenvalues. The eigenvalues λ_1 and λ_2 may be found by solving the quadratic equation below for x:

$$(a-x)(d-x) - bc = 0$$

Each of the solutions (λ_1 and λ_2) is related to an eigenvector $\vec{W} = \begin{pmatrix} u \\ v \end{pmatrix}$, which may be found by solving:

$$\mathbf{M} \cdot \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \cdot \begin{pmatrix} u \\ v \end{pmatrix}$$

(where * represents matrix multiplication) Next, each eigenvector is normalized by dividing it by its length $\sqrt{u^2 + v^2}$.

Two projection operators **P** and **Q** may now be constructed, each represented in terms of the corresponding normalized eigenvector $\begin{matrix} u \\ v \end{matrix}$:

$$\mathbf{P} = \begin{pmatrix} u \cdot u & v \cdot u \\ v \cdot u & v \cdot v \end{pmatrix}$$

Q is determined in the same way, using the second eigenvector values.

The result is two eigenvalues, λ_1 and λ_2, and two projection matrices, **P** and **Q**, and the desired

$$\mathbf{M_1} = \lambda_{1.} \cdot P + \lambda_{2.} \cdot Q$$

P represents one direction of scaling, and the amount of scaling is λ_1. **Q** represents the direction of scaling orthogonal to the direction of P_1, and the amount of scaling in that direction is λ_2.

Visually, applying $\mathbf{M_1}$ results in different scaling in each of the two directions, stretching in a given direction and different stretching/shrinking in the orthogonal direction.

The user can select one direction (bigger or smaller) by checking a checkbox) and force the other (orthogonal) scaling equal to 1 (i.e. no scaling). $\mathbf{M_0}$ is replaced by $P + \lambda_2 \cdot Q$ or by $\lambda_1 \cdot P + Q\lambda_1$***P** + **Q**, depending on which direction is chosen. A natural choice would be the direction of greater stretching.

This also gives an alternative approach to the extraction of the magnification component above: both λ_1 and λ_2 are replaced by

$\sqrt{\det}$ which is simply $\sqrt{\lambda_1 \cdot \lambda_2}$

If there is rotation or skew (i.e. the matrix is not symmetric) one choice is to extract the rotation first and then do the above. Alternatively, the skew may be extracted first, as described below.

In the first instance, we first extract the rotation. The rotation matrix $\mathbf{M_r}$ (from EQUATION 2 above) is first determined, and then its inverse calculated. Applying the inverse extracts out the rotation, and now the matrix will be symmetric and the above procedure may be applied. In the end, multiplying by **R** restores the rotation.

The result is similar if skew is extracted (as described below) instead of rotation. One is faced with two choices: (i) is it rotation or skew that is of interest? And (ii) is it horizontal or vertical skew that is of interest?

7. Extracting skew.

The inverse of the horizontal skew matrix is denoted $\mathbf{S^{-1}_H}$.

$$\mathbf{S_H} = \begin{pmatrix} 1 & -s \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{S^{-1}_H} = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

where

$$s = \frac{(c-b)}{d}$$

The inverse of the vertical skew matrix is denoted S^{-1}

The vertical skew matrix is

$$\mathbf{S_V} = \begin{pmatrix} 1 & 0 \\ -s & 1 \end{pmatrix}$$

$$\mathbf{S^{-1}_V} = \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}$$

where

$$s = \frac{(b-c)}{a}$$

Removing skew (horizontal or vertical) is done using the same approach as in the section above on removing rotation. Applying the inverse skew matrix removes the skew. In the end, the skew

matrix S is again applied to restore the skew and get the original matrix with the skew component removed.

8. Applying the Selected Component(s).

Photoshop enables a user to manipulate an electronic image with an onscreen "brush" controlled by an input device such as a mouse. Having extracted the desired component(s) of interest of the local distortion, the extracted component(s) may be applied using such a brush. The selected component(s) of the distortion can be applied under a brush cursor as the user moves the mouse, continuing to do so while a mouse button is pushed. As a result, the affine transform extracted from the initially selected starting point gets applied or "spread" everywhere the mouse goes while the mouse button is pushed.

That application need not be done all at once. It may be done gradually, in small steps. Also, application of the component under the cursor may be stronger (or faster) in the some areas and weaker in others. For example, the simplest implementation is circular brush shape with a circularly symmetric application function, which has a strength of 1 (i.e. fully applies the affine transform) at the center and decays to zero at the edges. Other "shapes" and weighting approaches can be used. For example, the brush shape need not be circular, or a multi-pole brush may be used in which a portion of the cursor applies one component, and another portion of the cursor applies a second component, which could be not only a warping effect, but other effects such as color blending.

Assume the paintbrush is currently at a particular location. The grid is repeatedly scanned under the brush a number of times per second, 30 times per second in one embodiment, and at each grid point under the brush, the current distortion D_0 at a point is modified to obtain a new value

$$D_{new} = (D_{final} - D)(scalar)$$

Here D_{final} is what the affine distortion is at this point as calculated in equation 1 above and the scalar is the value of the brush weight function.

D_{final} is given by EQUATION 1, where the Displacement at point A is the distortion at the initial click point, and $\vec{V}$ is the vector starting from that initial point and ending at the current

point under the paintbrush. This vector now could be very big if the user has moved the mouse far from the starting point. as opposed to $\vec{V}$ during extraction of the affine transform at the first click, when it was small.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, a variety of surrounding points may be used, different ways of averaging out the calculated affine transform, as well as different numbers of surrounding points may be used. Other approaches may be used for calculating the affine transform. Components may be applied in whole or in part or in various combinations. The affine transform may be additively or subtractively applied, inverted or scaled. Accordingly, other embodiments are within the scope of the following claims.